# Lecture 13: Uncertainty - 4

Victor Lesser

CMPSCI 683
Fall 2004

# Outline

- Approximate inference techniques -- chapter 14.4&14.5
- Alternative approaches to uncertain reasoning (will do later)
  - **Dempster-shafer**
  - **Fuzzy-Logic**
  - **Truth-maintenance**



# Mid Term Exam

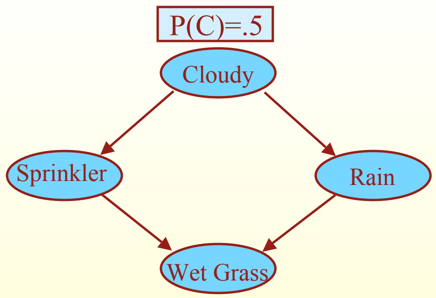
- **November 2 -- Tuesday; in class**
- **Open book but no computers**
- **Covering only material through chapter 14.5**
  - No material on utility theory or decision trees
- **Style of questions**
  - Mix of Short essay and Technique
  - Homework 3 is a good example



# Inference in Multiply Connected BNs

- Clustering **methods transform the network into a probabilistically equivalent polytree.**
  - Also called Join tree algorithms
- Conditioning **methods instantiate certain variables and evaluate a polytree for each possible instantiation.**
- Stochastic simulation **approximate the beliefs by generating a large number of concrete models that are consistent with the evidence and CPTs.**

## Example of Multiply Connected BN




| C | P(S) |
|---|---|
| T | .10 |
| F | .50 |

| C | P(R) |
|---|---|
| T | .80 |
| F | .20 |

| S | R | P(W) |
|---|---|---|
| T | T | .99 |
| T | F | .90 |
| F | T | .90 |
| F | F | .00 |

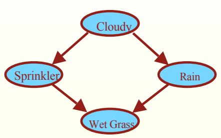


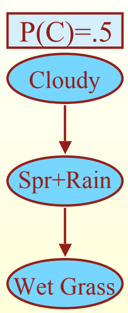
## Clustering Methods

- *Creating meganodes until the network becomes a polytree.*
- **Most effective approach for exact evaluation of multiply connected BNs.**
- **The tricky part is choosing the right meganodes.**
- **Q. What happens to the NP-hardness of the inference problem?**



## Clustering Example




| C | P(S+R) TT | TF | FT | FF |
|---|---|---|---|---|
| T | .08 | .02 | .72 | .18 |
| F | .10 | .40 | .10 | .40 |

| S+R | P(W) |
|---|---|
| T T | .99 |
| T F | .90 |
| F T | .90 |
| F F | .00 |

How do you still answer
P(Rain=True / Wet Grass=False) ?

What the disadvantages?

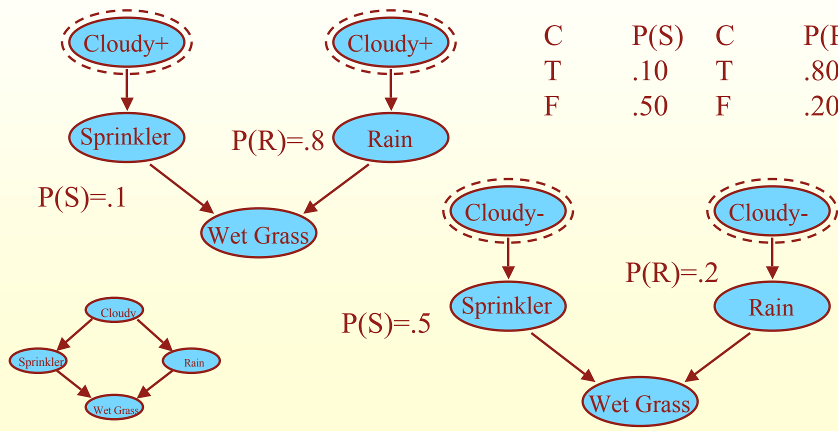


## Cutset Conditioning Methods

- *Once a variable is instantiated it can be duplicated and thus “break” a cycle.*
- **A cutset is a set of variables whose instantiation makes the graph a polytree.**
- **Each polytree’s likelihood is used as a weight when combining the results.**
- **Bounded cutset conditioning is an anytime version.**

## Networks Created by Instantiation

- Eliminate Cloudy from BN; Sum(Cloudy+,Cloud-)




| C | P(S) | C | P(R) |
|---|---|---|---|
| T | .10 | T | .80 |
| F | .50 | F | .20 |



## Stochastic Simulation

Direct Sampling

- **Assign each root node a value based on prior probability.**
- **Assign all other nodes a NULL "value".**
- **Pick a node X with no value, but whose parents have values, and randomly assign a value to X**
  - using P(X|Parents(X)) as the distribution. Repeat until there is no such X.
- **After N trials, P(X|E) can be estimated by occurrences(X and E) / occurrences(E).**



## Stochastic Simulation cont.

- **Problem with very unlikely events.**
- **Likelihood weighting can be used to fix problem.**
- **Likelihood weighting converges much faster than logic sampling and works well for very large networks.**

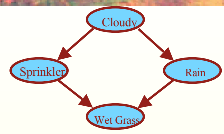


## The MCMC Algorithm

- **MCMC generates each event by making a random change to the preceding event.**
  - It is therefore helpful to think of the network being in a particular *current state* specifying a value for every variable.
- **The next state is generated by randomly sampling a value for one of the nonevidence variables $X_i$, *conditioned on the current values of the variables in the Markov blanket of $X_i$.***
  - MCMC therefore wanders randomly around the state space—the space of possible complete assignments—flipping one variable at a time but keeping the evidence variables fixed.

## Example of Likelihood Weighty

*P*(WetGrass/Rain)




- Choose a value for *Cloudy* with prior *P*(*Cloudy*) = 0.5. Assume we choose *cloudy* = *false*.
- Choose a value for *Sprinkler*. We see that *P(Sprinkler* | ¬ *Cloudy*) = 0.5, so we randomly choose a value given that distribution. Assume we choose *Sprinkler* =*True*.
- Look at *Rain*. This is an evidence variable that has been set to *True*, so we look at the table to see that *P*(*Rain* | ¬ *Cloudy*) = 0.2. This run therefore counts as 0.2 of a complete run.



## Example of Likelihood Weighty cont'd

- **Look at *WetGrass*. Choose randomly with *P*(*WetGrass* | *Sprinkler* ∧*Rain*) =0.99; assume we choose *WetGrass* = *True*.**
- **We now have completed a run with likelihood 0.2 that says *WetGrass* = *True* given *Rain* = *True.* The next run will result in a different likelihood, and (possibly) a different value for *WetGrass*. We continue until we have accumulated enough runs, and then add up the evidence for each value, weighted by the likelihood score.**

**Likelihood weighting usually converges much faster than logic sampling**

**Still takes a long time to reach accurate probabilities for unlikely events**



## Summary of a Belief Networks

- Conditional independence **information is a vital and robust way to structure information about an uncertain domain.**
- Belief networks **are a natural way to represent conditional independence information.**
  - The links between nodes represent the qualitative aspects of the domain, and the conditional probability tables represent the quantitative aspects.
- **A belief network is a complete representation for the joint probability distribution for the domain, but is often exponentially smaller in size.**



## Summary of a Belief Networks, cont'd

- **Inference in belief networks means computing the probability distribution of a set of query variables, given a set of evidence variables.**
- **Belief networks can reason causally, diagnostically, in mixed mode, or intercausally. No other uncertain reasoning mechanism can handle all these modes.**
- **The complexity of belief network inference depends on the network structure. In** polytrees **(singly connected networks), the computation time is linear in the size of the network.**

## Summary of a Belief Networks, cont'd

- **There are various inference techniques for general belief networks, all of which have exponential complexity in the worst case.**
  - In real domains, the local structure tends to make things more feasible, but care is needed to construct a tractable network with more than a hundred nodes.
- **It is also possible to use approximation techniques, including** stochastic simulation, **to get an estimate of the true probabilities with less computation.**



## Making Simple One-Shot Decisions

- **Combining Beliefs and Desires Under Uncertainty**
- **Basis of Utility Theory**



## Maximum Expected Utility (MEU)

- The MEU principle says that a rational agent should choose an action that maximizes its expected utility in the current state (E)

$EU(\alpha|E) = \max_A \sum_i P(Result_i(A)|Do(A),E)\ U(Result_i(A))$

- Why isn't the MEU principle all we need in order to build "intelligent agents"?
  - **Is it Difficult to Computer P,E or U ?**



## MEU Computational Difficulties

- Knowing the current state of the world requires perception, learning, knowledge representation and inference.
- Computing P(*) requires a complete causal model of the world.
- Computing the utility of a state often requires search or planning (distinguish between explicit and implicit utility)
  - Calculation of Utility of a particular state may require us to look at what utilities could be achieved from that state
- All of the above can be computationally intractable, hence one needs to distinguish between "perfect rationality" and "resource-bounded rationality" or "bounded-optimality".
- Also Need to consider more than one action (one-shot decisions versus sequential decisions).

# Next Lecture

- **Making decisions under uncertainty using utility theory. --chapter 16**
  - The value of information.

- Decision Trees